Massachusetts Institute of Technology
C. S. Draper Laboratory
Cambridge, Massachusetts

COLOSSUS Memo #310, LUMINARY Memo #201

To: Distribution

From: D. Densmore

Date: 16 February 1971

Subject: DOWNRUPT SEQUENCING, LOST DOWNRUPTS, AND IMPAIRED DOWNLINK INFORMATION

The following is an analysis of the interactions of the AGC software down-telemetry program, the hardware priority interrupt system, and the downlink telemetry hardware, with consideration to how abnormal sequencing affects the downlists received by the ground. The list provided to the RTCC in Houston via radio telemetry and the list printed out on simulation downlink edits are contrasted. Their purposes are different, and under some conditions the contents of the lists would be different.

Down telemetry is transmitted at a high bit rate of 50 frames per second, including two AGC words per frame, or a low bit rate of 10 frames per second. This discussion is based on timing with the high bit rate. (A frame consists of 128 eight-bit telemetry words, 5 of which are used for AGC information. These 40 bits provide the information in channels 34 and 35 along with a word order code.)

1. HOW THE DOWN TELEMETRY PROGRAM WORKS.

The down telemetry program is initiated by an interrupt which is requested every 20 milliseconds. It selects the appropriate word of the designated fixed-memory downlink list, picks up the two AGC words indicated out of erasable memory, and loads those two words into channels 34 and 35. It also sets bit 7 of channel 13 to "0" to flag the first (ID) and 51st (AGC time) words of the list. For words 2-50 and 52-100 this bit (called the word order code) is set to one.

The word order code is tested at the beginning of the downlink program processing. If it is "0" (as it would be if this were the pass for either word 2 or word 52 of the list) then it is set to "1" (with the necessary call to C13STALL before writing into channel 13). If it is found to be "1" already no change is made before the downlink processing is entered to select the next downlist word.

If the program finds that a list was completed on the last pass or that the internal pointers indicate that the next word on the list is AGC time, then before writing the ID or the value of time from erasable into channels 34 and 35 the program writes zero into the word order code, preceded of course by a call to C13STALL. The downlink erasable dump sets the word order code to zero when a special ID is sent to flag the beginning of each new erasable bank.

In the case of these two words on the downlist, then, channel 13 is written into just before channels 34 and 35 are. In the case of the two words that immediately follow (words 2 and 52) it is done some milliseconds earlier. By an unfortunate coincidence, the 2nd and 52nd words of the downlists are "snapshot" passes, which store the last twelve registers of state vector information into a buffer before placing the first two in channels 34 and 35. While a normal downrupt processing pass takes only about 1 millisecond, the snapshot passes take 3.6 ms. A related problem is that the call to C13STALL could possibly cause a further delay in the interrupt of 5 milliseconds. Therefore there is a possibility that the writing into channel 34 and 35 will come too late for transmission with the new word order code in channel 13, since the contents of the channels are read out and transmitted to the ground on a regular basis which is not dependent on when the channels are written into.

2. HOW THE INTERRUPT PRIORITY CONTROL AND THE DOWNLINK TELEMETRY CONVERTER WORK.

The telemetry hardware works on a 20 ms cycle. Every 20 ms a "start pulse" is generated which begins a process of strobing out the contents of channels 34 and 35 plus bit 7 of channel 13. This strobing process takes

.8 milliseconds. At the end an "end pulse" is generated which is a request signal for a program interrupt. The downlink telemetry converter meanwhile generates a 40-bit word from the contents on channels 13, 34 and 35. This 40-bit word (5 8-bit "telemetry words") is transmitted serially. The word order code is the leading bit, followed by the contents of channels 34 and 35, and then the last 7 bits of channel 34 are repeated to fill up the 40-bit transmission word.

The request signal for the downrupt is routed to the downrupt request flip flop. "The outputs of the flip flops are interconnected in such a manner that if a higher priority request is present, the outputs of all lower priority request flip-flops are inhibited." [1] Interrupt requests for T6RUPT, T5RUPT, T3RUPT, T4RUPT, KEYRUPT1, KEYRUPT2, and UPRUPT are of higher priority than downrupts. A downrupt request must wait until all waiting requests of higher priority have been honored. When the downrupt is the highest priority request (and provided that if it was requested while an INHINT was active a RELINT had been executed) it will be honored by a transfer of control to location 4040, the down telemetry program interrupt lead-in. This resets the downrupt request flipflop, which can now register another downrupt request. Once the flip-flop has been set indicating one downrupt request waiting, subsequent requests will have no effect. This is what results in downrupts getting "lost" in real life.*

The "start pulse" of the downlink telemetry converter, then, initiates the strobing out of the information in the channels. The "end pulse" generates a downrupt request. The downrupt request signal is generated every 20 ms, and 19.2 ms later the contents of the channels are strobed out in a process that takes .8 ms. The timing on the downlink program processing, however, depends on what INHINTS are active and what higher priority interrupts are waiting or come in before the downrupt request is honored. If there is too long a delay the validity of the downlink information transmitted to the ground will be impaired in one of three ways (see Section 4).

3. SIMULATOR LOST DOWNRUPT MESSAGE

When a "DOWNRUPT LOST" message appears on a simulation, this means a particular case in the set of real-life contingencies that can result in impaired downlink information. The Simulator message occurs when an n^{th} downrupt request has been made before the $(n-1)^{th}$ has been honored. This would occur approximately 20 milliseconds after the $(n-1)^{th}$ downrupt was called for by the telemetry end-pulse.

The Simulator prints a message if a downrupt request is not honored within 20 ms. Actually the information may be lost or impaired if it is delayed longer than about 15.5 ms (this number allows maximum downlink processing time before the channel read-out begins).

The Simulator recognizes the loss of a downrupt request. In this case if a trace is put on location 4040 it would be noticed that the gap between two of the downrupt processing initiations is approximately 40 ms instead of the usual 20 ms.

If one checks the downlink edit associated with the simulation it would be seen that no processing had been lost, since all the words are there on the list for the period containing the "lost downrupt". However, the edit does not accurately model the list transmitted to the ground by the downlink telemetry hardware. No attempt is made by the Simulator to model the telemetry hardware "start pulse". The dumping of the channels for the edit takes place not 19.2 or even 20 ms after the end-pulse but rather when the channels are loaded by the down-telemetry program. The downlink edit was not designed for investigation of lost downrupts. It checks that the downlists are encoded properly and that the downlink program works.

One would be able to observe, however, that that particular 100-word list took 2.02 seconds to accumulate on the MARSROT instead of the normal 2 seconds. This information can be found in the octal-only dumps by comparing the AGC time (word 51) in successive downlists. The time

period in which the downrupt was lost should be 20 ms longer. In the engineering edit the list following the list during which the downrupt was lost will say it covers a period beginning 2.02 seconds after the previous instead of the usual 2 seconds.

It may be possible to make the downlink edit somewhat more useful and accurate by requesting that the channels be dumped every 20 ms instead of on location TMRESUME. For this to work the dump would have to be properly synched with the initiation of the end-pulse simulation (called for by the TELEM2 request). The best arrangement would be to dump the channels 19.2 ms after the downrupt request. Alternatively the edit could be changed to take the time the channels were written into and compare it to the time of the simulated end-pulse, if that information could be made available to the edit. Messages could then be printed on the edit about lost or questionable information.

The situation recognized by the Simulator as a "lost downrupt", then, is the case where a downrupt request is overwritten before being honored. In simulation edits this downrupt appears to have vanished without a trace except for 20 milliseconds added to the transmission time of the list. The following section notes what would happen if this timing occurred in flight.

4. IMPAIRMENT IN AGC DOWNLINK INFORMATION TRANSMITTED TO THE GROUND BY THE DOWNLINK TELEMETRY HARDWARE

There are three conditions that can be produced by delayed honoring of downrupt requests.

The exact Δt which will cause the problem depends on the duration of that particular pass through the downlink program. As indicated above a problem could arise if the downrupt request is not honored within 15.5 ms.

In the third case the end of the downlink program pass (where the channels are written) is delayed so that the channels are being written in the strobe-out window 19.2 - 20 ms after the downrupt request was made by the start pulse.

In the second case the end of the program pass is delayed longer than 20 ms, although the request is honored before 20 ms.

In the first case the honoring of the request is delayed longer than 20 ms.

CASE 1.

The first is the correlate of the Simulator "downrupt lost". However, because the Simulator does not simulate the start pulse, the downlink edit does not accurately model the consequences in real life of this case. Simulation edits show normal lists in this case, with the usual 100 words in the proper order, anomalous only in that the "transmission" time is 20 ms longer for each downrupt lost.

The start pulses of the downlink telemetry hardware come in periodically to strobe out the downlink channel contents and transmit the information to the ground. The case here is that downrupt request n is waiting to be honored, but it is inhibited by higher priority interrupt requests. 19.2 milliseconds after the request signal went out the next start pulse will begin reading out the channels. Under normal conditions it would be picking up downlink information loaded by the downlink processing triggered by downrupt n. In this case it reads out the information left in the channels from downrupt n-1 processing. The result is a repeated word on the downlink. When this information is strobed out (again), downrupt n+1 is requested by the end-pulse. This request has no effect, however, since the downrupt flip-flop is still set by request n. If the waiting request is honored within a reasonable time, the downlink program places the next word in the channels and the start pulse strobes that out for transmission.

The result for the ground is that this list takes 2.02 seconds to transmit and has one extra (repeated) word.

CASE 2.

The second situation, not recognized by the Simulator, occurs if downrupt request n+1 is signalled by an end-pulse after request n was honored (control transferred to location 4040, the interrupt lead-in) but before processing triggered by downrupt n had loaded new data into the down-link channels. To state it more precisely this situation occurs if the downrupt request n is delayed so long that the next start-pulse strobes out the downlink channels before the processing finally triggered by that downrupt has written new information, but not so long that the end-pulse that signals downrupt request n+1 occurs before request n has been honored.

The processing triggered by downrupt n will put the new information in the channels, too late for transmission by the start-pulse that would normally have picked it up. Downrupt n+1 will then come in and (provided of course it is not also delayed unreasonably) write over the information supplied by processing for n. The start-pulse will then pick up the information supplied by processing of downrupt n+1.

The result for the ground is again that word n-1 is repeated, but this time no interrupt request has been lost, therefore the list will be transmitted in the normal 2 seconds. The repeated word is on the list <u>instead</u> of the word that should have been in that position on the list. That n^{th} word is lost since it was provided in the wrong time-slot between start-pulses and was overwritten by the <u>next</u> word. The next $(n+1)^{th}$ word is correct and appears in the right position.

CASE 3.

In the third case, which again is not recognized by the Simulator, downrupt n is delayed so long that it is actually writing into the downlink channels while they are being strobed out by the following start-pulse. The word that is transmitted is neither a repetition of word n-1 nor the proper word n. The information that would have been supplied as a result of the processing of downrupt n is lost. When downrupt n+1 comes in (requested by the end-pulse .8 ms after the start pulse that got the bad word) it will (under normal timing) supply the next word in the list as if the previous word had gone out properly.

The result for the ground here is a bad word for word n and the rest of the list normal.

In case 1, if the RTCC finds more than 50 words in the first half of the list (i. e. the 51st word is not word-order-code-zero Time 1 Time 2) it throws away the whole list and begins again with the next word-order-code-zero ID word. If it finds more than 50 words in the second half of the list (the 101st word isn't word-order-code-zero ID) it throws away just the second half of the list.

Cases 2 and 3 are the same to the RTCC in the sense that the "repeated" or "bad" word is in either case a "garbage" word compared to what should have been there on the list; and the word would probably appear as "garbage" when it is scaled and displayed.

Reference

[1] AC Apollo G&N Study Guide

*This corrects the erroneous statement in Luminary Memo #89, A Metaphysic of Downrupts, that downrupts themselves are not actually lost.

The information in this paragraph is taken from the document cited in reference [1].